

Project - Network Simulator

Swastik Gupta(BITE013),Manish Kumar(BITE043),Gurveen Singh(BITE005)

0.1 Objective:

Demonstrating the working of all the layers of Network Protocol Stack using Network Simulator

0.2 Programming Language used:

C++

0.3 Details:

- Object oriented programming concepts like creation of classes and objects is used.
 - Class **Device** - MACaddress , data , setMACaddress and seddata are the attributes.The objects created of this class are the nodes in the network.
 - Class **Hub** - data2, macAdd ,store and hub2dev are the attributes.The object created (hb1) keeps a check on whether data is reviewed at the correct device or not from the sender.
- The simulation in this project shows the flow of data from one node to other.
- Firstly, the user has been given the option to choose the required network layer . After that , the user is expected to choose the source device and the destination device.

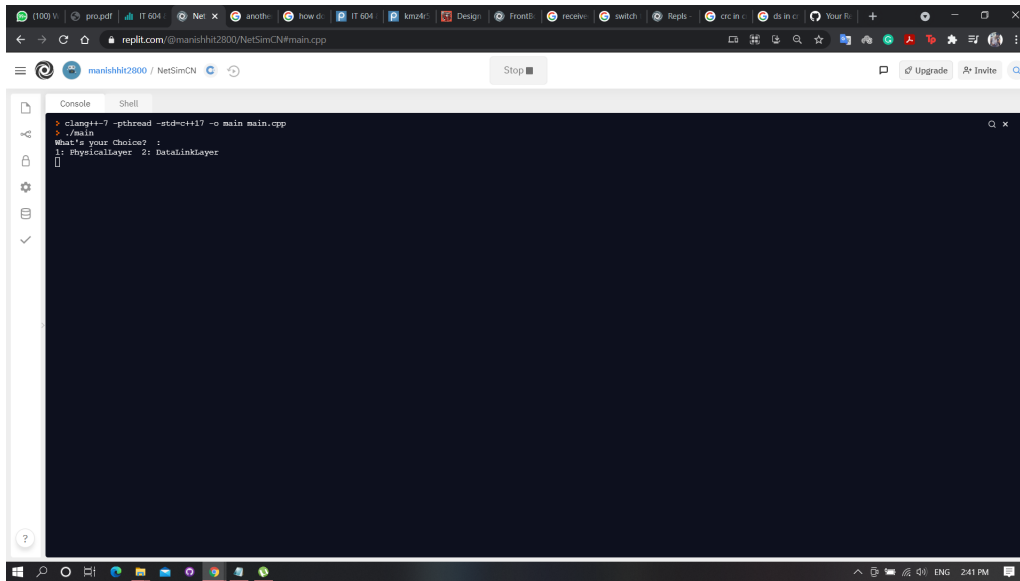
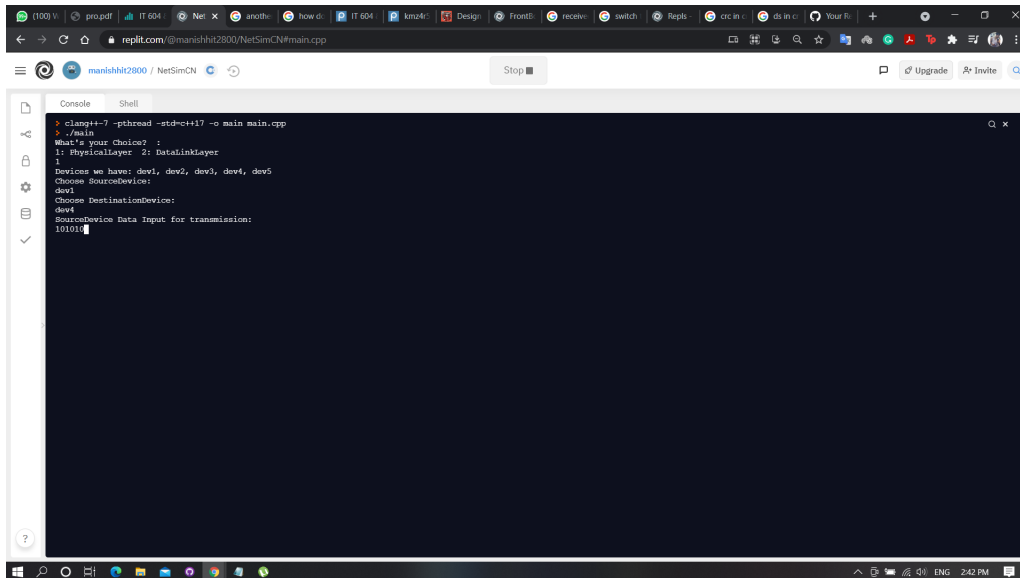


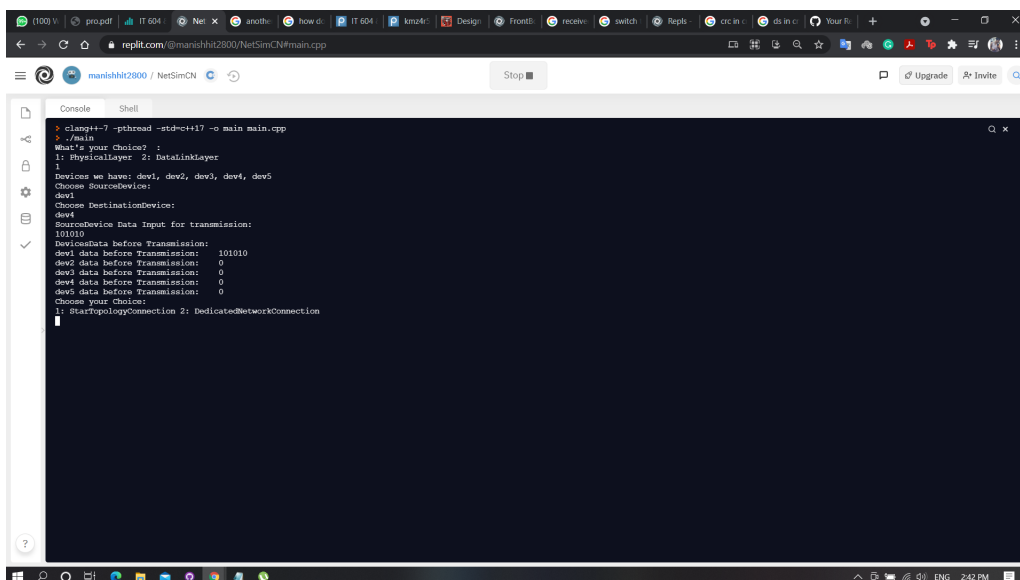
Figure 1: User being given the option to choose the layer

- Once this is done , the user is asked for data he/she wants to transfer. The data is transferred in the form of bits.
- Then the data has been transferred with the help of functionalities of the selected layer.



```
clang++-7 -pthread -std=c++17 -o main main.cpp
./main
What's your Choice? :
1: PhysicalLayer 2: DataLinkLayer
1
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
101010
```

Figure 2: User is expected to enter the input



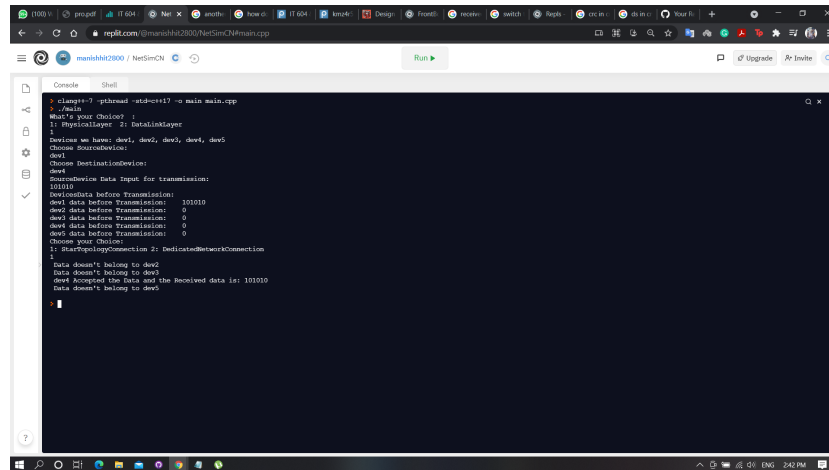
```
101010
DeviceData before Transmission:
dev1 data before Transmission: 101010
dev2 data before Transmission: 0
dev3 data before Transmission: 0
dev4 data before Transmission: 0
dev5 data before Transmission: 0
Choose your Choice:
1: StartTopologyConnection 2: DedicatedNetworkConnection
```

Figure 3: Data each device contains before the transmission

0.4 Functionlities of the Physical layer

Transmission of bits from one device to other. The user is given two options for the transmission of data.They are -

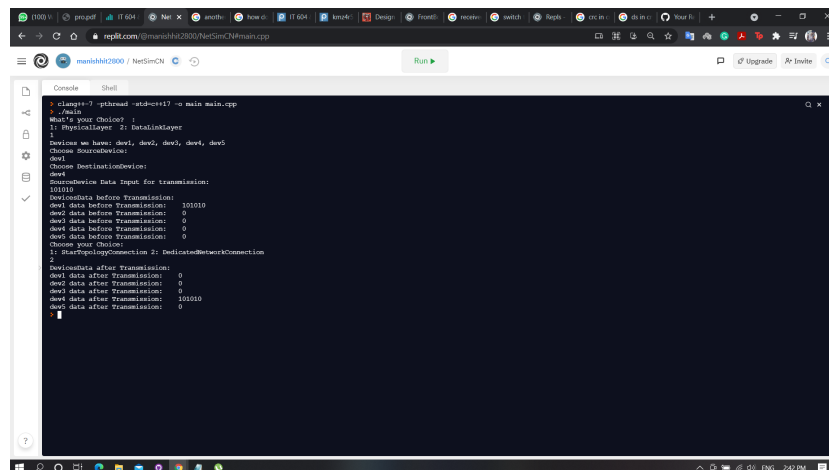
- Using a Dedicated network connection



```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
What's your Choice: 1
1: PhysicalLayer 2: DataLinkLayer
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
101010
Deviondata before Transmission:
dev1 data before Transmission: 101010
dev2 data before Transmission: 0
dev3 data before Transmission: 0
dev4 data before Transmission: 0
dev5 data before Transmission: 0
Choose your Choice:
1: StarTopologyConnection 2: DedicatedNetworkConnection
1
Data doesn't belong to dev2
Data doesn't belong to dev3
dev4 Accepted the data and the Received data is: 101010
Data doesn't belong to dev5
```

Figure 4: Dedicated network

- Using a Star topology connection

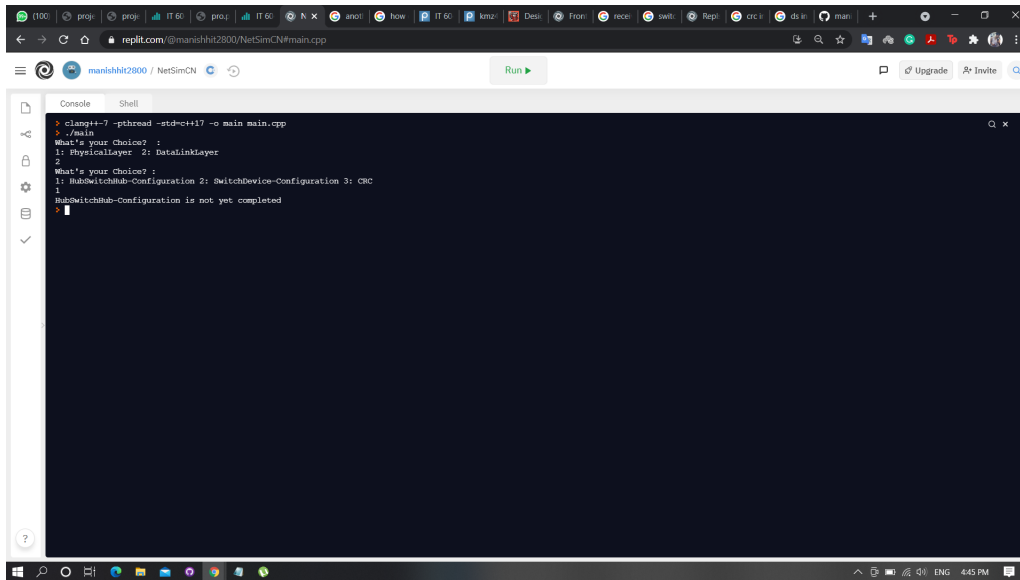


```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
What's your Choice: 1
1: PhysicalLayer 2: DataLinkLayer
Devices we have: dev1, dev2, dev3, dev4, dev5
Choose SourceDevice:
dev1
Choose DestinationDevice:
dev4
SourceDevice Data Input for transmission:
101010
Deviondata before Transmission:
dev1 data before Transmission: 101010
dev2 data before Transmission: 0
dev3 data before Transmission: 0
dev4 data before Transmission: 0
dev5 data before Transmission: 0
Choose your Choice:
1: StarTopologyConnection 2: DedicatedNetworkConnection
2
Deviondata after Transmission:
dev1 data after Transmission: 0
dev2 data after Transmission: 0
dev3 data after Transmission: 0
dev4 data after Transmission: 101010
dev5 data after Transmission: 0
```

Figure 5: Star topology

0.5 Functionlities of the Data link layer

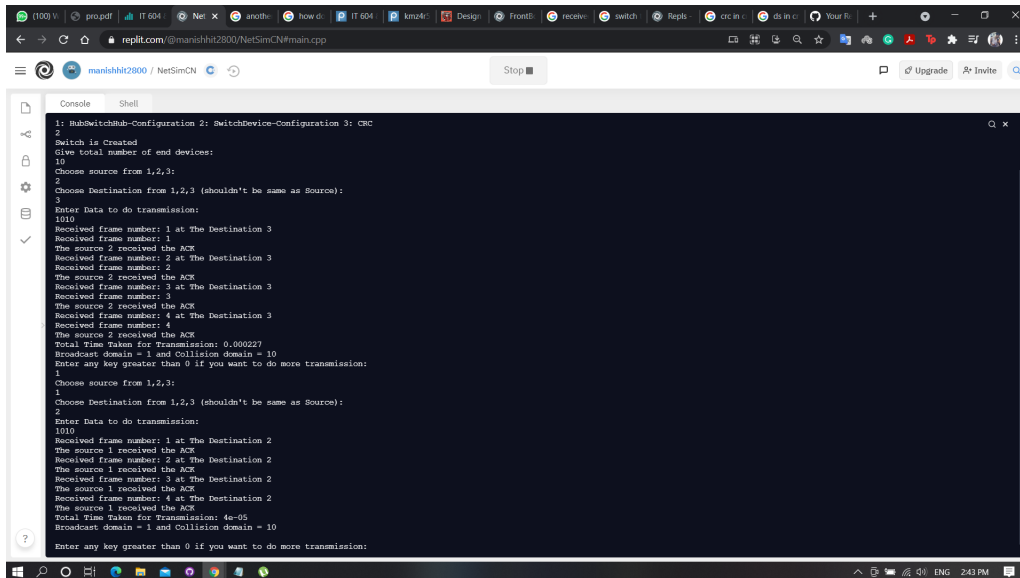
- Hub switch configuration



```
Clang++-7 -pthread -std=c++17 -o main main.cpp
./main
What's your Choice? :
1: PhysicalLayer 2: DataLinkLayer
2:
What's your Choice? :
1: HubSwitchHub-Configuration 2: SwitchDevice-Configuration 3: CRC
1:
HubSwitchHub-Configuration is not yet completed
```

Figure 6: Hub switch configuration is under construction and not yet complete

- **Switch device configuration** - In this configuration , switch is being created and the user is asked to enter the total number of end devices.Also once the reciever recieves the frame, it send back an ACK to the source.
Total time for transmission is shown after the transmission is complete. The user is then asked if he/she wants to more transmission.



```
1: HubSwitchHub-Configuration 2: SwitchDevice-Configuration 3: CRC
2
Switch is Created
Give total number of end devices:
10
Choose source from 1,2,3:
2
Choose Destination from 1,2,3 (shouldn't be same as Source):
3
Enter Data to do transmission:
1010
Received frame number: 1 at The Destination 3
The source 2 received the ACK
Received frame number: 2 at The Destination 3
The source 2 received the ACK
Received frame number: 3 at The Destination 3
The source 2 received the ACK
Received frame number: 4 at The Destination 3
The source 2 received the ACK
Total Time Taken for Transmission: 0.000227
Broadcast domain = 1 and Collision domain = 10
Enter any key greater than 0 if you want to do more transmission:
1
Choose source from 1,2,3:
1
Choose Destination from 1,2,3 (shouldn't be same as Source):
2
Enter Data to do transmission:
1010
Received frame number: 1 at The Destination 2
The source 1 received the ACK
Received frame number: 2 at The Destination 2
The source 1 received the ACK
Received frame number: 3 at The Destination 2
The source 1 received the ACK
Received frame number: 4 at The Destination 2
The source 1 received the ACK
Total Time Taken for Transmission: 4e-05
Broadcast domain = 1 and Collision domain = 10
Enter any key greater than 0 if you want to do more transmission:
```

Figure 7: Functioning of switch device configuration

- **Error Control** - Achieved by using the concept of **CRC** .
The user is asked to enter the source input and the coefficients of the generator polynomial.
After this, the data to be sent is calculated.
The user is then expected to enter the recieved data and therefore the remainder is calculated.
If the remainder is in the form of string of 0's, there is no error.
If there is atleast one 1 , there is a error.
We have shown below two test cases for error detection.

The screenshot shows a C++ program running in a terminal window on Replit. The program is a CRC error detection tool. It prompts the user for their choice of operation (1: PhysicalLayer, 2: DataLinkLayer). The user enters 2. Then, it prompts for the source input (1100) and the coefficients of the generator polynomial (1101). It calculates the dividend (1100000) and the data to send (1100101). The user enters the received data (1100101). The program calculates the remainder (000) and displays the message "Correct Data Received Without Any Error".

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
./main
What's your Choice? :
1: PhysicalLayer 2: DataLinkLayer
2
What's your Choice? :
1: HubSwitchHub-Configuration 2: SwitchDevice-Configuration 3: CRC
3
Enter Source Input:
1100
Enter Coefficients of generator polynomial:
1101
Input: 1100
Polynomial: 1101
Divident: 1100000
Data to send: 1100101
Enter Data Received:
1100101
Remainder is: 000
Correct Data Received Without Any Error
```

The screenshot shows the same C++ program running in a terminal window on Replit. The user enters 2 for the choice of operation. Then, it prompts for the source input (1100) and the coefficients of the generator polynomial (1101). It calculates the dividend (1100000) and the data to send (1100101). The user enters the received data (1100100). The program calculates the remainder (001) and displays the message "Data Received Contain Some Error".

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
./main
What's your Choice? :
1: PhysicalLayer 2: DataLinkLayer
2
What's your Choice? :
1: HubSwitchHub-Configuration 2: SwitchDevice-Configuration 3: CRC
3
Enter Source Input:
1100
Enter Coefficients of generator polynomial:
1101
Input: 1100
Polynomial: 1101
Divident: 1100000
Data to send: 1100101
Enter Data Received:
1100100
Remainder is: 001
Data Received Contain Some Error
```

Figure 8: Test cases for error detection